

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

Jc971 U.S. PTO  
09/817259  
03/27/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2000年 3月28日

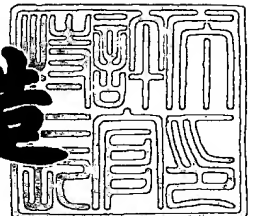
出 願 番 号  
Application Number: 特願2000-088703

出 願 人  
Applicant (s): 株式会社東芝

2000年 9月 1日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



出証番号 出証特2000-3070733

【書類名】 特許願

【整理番号】 A000000707

【提出日】 平成12年 3月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 13/00

【発明の名称】 自動分散処理システムおよび記録媒体

【請求項の数】 7

【発明者】

    【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝府中工場内

    【氏名】 村松 孝治

【特許出願人】

    【識別番号】 000003078

    【氏名又は名称】 株式会社 東芝

【代理人】

    【識別番号】 100058479

    【弁理士】

    【氏名又は名称】 鈴江 武彦

    【電話番号】 03-3502-3181

【選任した代理人】

    【識別番号】 100084618

    【弁理士】

    【氏名又は名称】 村松 貞男

【選任した代理人】

    【識別番号】 100068814

    【弁理士】

    【氏名又は名称】 坪井 淳

【選任した代理人】

    【識別番号】 100092196

    【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 自動分散処理システムおよび記録媒体

【特許請求の範囲】

【請求項 1】 命令委託元マシンと命令委託先マシンがネットワークを介して接続されている自動分散処理システムにおいて、

前記命令委託元マシンは、スレッド識別子のもとに各スレッドを管理するテーブルと、委託元上位ライブラリとともに委託元アプリケーションの処理過程における命令発生時、前記テーブルで管理するスレッド識別子を付加して命令を前記委託先マシンに送信する委託元命令中継スレッドと、前記委託先マシンからの別命令を処理するためのスレッドの振り分けを行う委託元命令振り分けスレッドとを有する命令中継ライブラリを設け、

前記命令委託先マシンは、前記委託元命令中継スレッドから送信されてくる命令をスレッド識別子とともに受信し、命令を処理するスレッドを作成し前記スレッド識別子とともに渡す委託先命令振り分けスレッドと、この受け取った命令を委託先上位ライブラリとともに処理し、その命令処理時の別命令発生や処理終了時、前記スレッド識別子を付加して前記別命令や処理終了返信を前記委託元命令振り分けスレッドに送信する命令処理スレッドとを有する命令実行モジュールを設けたことを特徴とする自動分散処理システム。

【請求項 2】 命令を委託する委託元マシンのコンピュータを動作させるためのプログラムを記録した記録媒体において、

前記プログラムは、

委託元アプリケーションの処理時に命令が発生したとき、テーブルで管理するスレッド識別子と各スレッドとの関係を参照し前記命令にスレッド識別子を付加して委託先マシンに送信する命令中継機能と、前記委託先マシンによる命令処理時に発生する別命令に前記スレッド識別子が付加されて送信されてきたとき、委託元となったスレッドに振り分けるスレッド振り分け機能と、この振り分けられた委託元スレッドで処理された前記別命令の処理結果を前記委託先マシンに返信する機能とを実現する前記コンピュータ読み取り可能な記録媒体。

【請求項 3】 委託元マシンから命令を委託される委託先マシンのコンピュ

ータを動作させるためのプログラムを記録した記録媒体において、

前記プログラムは、

前記委託元マシンからスレッド識別子とともに命令を受信したとき、このスレッド識別子のもとに命令を処理するスレッドを作成するスレッド作成機能と、このスレッド作成機能によって作成されたスレッドが前記命令の処理時に別命令が発生したとき、前記受信したスレッド識別子を付加して別命令を前記委託元マシンに送信する機能とを実現する前記コンピュータ読み取り可能な記録媒体。

【請求項4】 命令委託元マシンと命令委託先マシンがネットワークを介して接続されている自動分散処理システムにおいて、

前記命令委託元マシンおよび命令委託先マシンは、それぞれ排他用ロック後に自身のアプリケーション処理時に命令が発生したとき、ロックをかけて相手側マシンに命令を中継する命令中継スレッドおよびこの命令中継スレッドからの命令を受信し処理する命令処理スレッドが設けられ、

そのうち少なくとも前記委託先マシン側命令処理スレッドは、前記委託元マシン側の命令を受信後、自身のマシンにてロック取得可能か否かを判断し、ロック取得不可の場合には前記委託元マシンにリトライを通知する手段と、ロック取得可能な場合にはロックを取得し命令処理終了後に返信し当該ロックを解放する手段とを有し、

また、少なくとも前記委託元マシン側命令中継スレッドは、前記委託先マシン側からリトライ通知を受けたとき、前記ロックを一時的に解放した後再び取得して再度命令を中継するリトライを実施する手段と、前記委託元マシン側から命令終了の返信を受けたとき、前記ロックを解放する手段とを備えたことを特徴とする自動分散処理システム。

【請求項5】 命令を委託する委託元マシンのコンピュータを動作させるためのプログラムを記録した記録媒体において、

前記プログラムは、

排他用ロックを行って自身のアプリケーション処理時に命令が発生したとき、委託先マシンの命令処理スレッドに命令を中継し受信待ちに設定する命令中継機能と、この委託先マシンの命令処理スレッドから前記命令処理の際にリトライ通

知を受けたとき、前記ロックを一時的に解放した後再び取得して再度命令を中継するリトライを実施する命令中継リトライ機能と、前記委託先マシンの命令処理スレッドから命令終了の返信を受けたとき、前記ロックを解放し命令を終了する命令処理終了機能とを実現する前記コンピュータ読み取り可能な記録媒体。

【請求項 6】 委託元マシンから命令を委託される委託先マシンのコンピュータを動作させるためのプログラムを記録した記録媒体において、

前記プログラムは、

前記委託元マシンから命令を受信後、自身のマシンにてロック取得可能か否かを判断するロック取得判断機能と、この判断機能によってロック取得不可能と判断されたとき前記委託元マシンにリトライを通知するリトライ通知機能と、前記判断機能によってロック取得可能と判断されたとき排他用ロックを取得し受信した命令を処理し、その命令処理終了後に前記委託元マシン返信し当該ロックを解放する命令処理機能とを実現する前記コンピュータ読み取り可能な記録媒体。

【請求項 7】 命令委託元マシンとイベント処理機能をもつ命令委託先マシンがネットワークを介して接続されている自動分散処理システムにおいて、

前記命令委託元マシンは、排他用ロック後のアプリケーション処理時に第 1 の命令が発生したとき、その命令内容に応じて前記ロックを解放し前記第 1 の命令を前記委託先マシンの命令処理スレッドに送信する手段およびこの命令処理スレッドにおける命令処理の終了を受けて第 1 の命令を終了する手段とを有する委託元命令中継スレッドと、前記委託先マシンのイベント処理スレッドから送信されてくる第 2 の命令を処理する委託元命令処理スレッドとを備え、

前記命令委託先マシンは、前記命令中継スレッドから第 1 の命令を受信したとき、排他用ロックを行って当該第 1 の命令を処理し、ロックを解放した上でイベント処理スレッドからの再開依頼がなされるまで待機し、再開後、その命令処理の終了後にロックを解放し、前記命令処理終了を前記委託元命令中継スレッドに委託し、前記待機中の前記命令委託先マシンに再開依頼を通知する前記委託先命令処理スレッドと、排他用ロックを行って自身のイベント処理時に第 2 の命令が発生したとき、前記委託元命令処理スレッドに委託する前記委託先イベント処理スレッドとを備えたことを特徴とする自動分散処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、処理の分散化により発生するデッドロックを回避する自動分散処理システムおよび記録媒体に関する。

【0002】

【従来の技術】

一般に、自動分散処理システムは、本来単独のマシン上で実行されるべき処理の一部を他のマシンに委託し、処理の分散化を図るシステムである。

このような自動分散処理システムでは、本来単独のスレッドまたはプロセスにより行うべき処理が委託元と委託先とに分散されるので、排他的機能が働き、本来の動作とは異なるものとなってしまう、いわゆるデッドロックが発生する。

【0003】

以下、デッドロックの発生について図9を参照して説明する。

【0004】

今、委託元マシン1においてアプリケーションの実行時、命令中継スレッド（ユーザスレッド）131が委託先マシン2に委託するための命令が発生すると、その命令を委託先マシン2に中継する。従って、委託元マシン1は、この段階では図示点線矢印のごとく委託処理待ちの状態となる。

【0005】

一方、委託先マシン2の命令処理スレッド2123は、委託元マシン1から委託される命令を受信し、その命令を処理するが、この命令処理の過程で別の命令が発生し、委託元マシン1の命令処理スレッド133に委託する必要があるとき、その別の命令を委託元マシン1に送信する。従って、委託先マシン2は、この段階では図示点線矢印のごとく委託処理待ちの状態となる。つまり、両マシン1、2は何れも命令処理待ちの状態となっている。

【0006】

しかし、別の命令を受信した委託元マシン1では、既に自身のアプリケーション処理時に排他用ロックをかけた状態となっており、待ちの状態にあるので、委

託先マシン 2 から委託した別の命令を処理できない状態にあり、委託元マシン 1 ではデッドロックが発生する。つまり、両マシン 1, 2 とも委託された命令を処理できない状態が発生する。

【 0 0 0 7 】

【発明が解決しようとする課題】

従って、以上のような自動分散処理システムにおいては、次のような問題が生ずる。

【 0 0 0 8 】

その 1 つは、委託先マシン 2 の処理部から排他用処理部分を除去することが考えられるが、それでは委託先マシン 2 の処理部をそのまま生かしつつ使用することができない問題がある。また、委託先マシン 2 の処理部の実装内容をマシン開発者から取得した上、排他部分を回避する考えもあるが、常にマシン開発者から実装内容を取得できるとは限らず、仮に取得したとしても排他機能を完全に回避できるとは言い難い。

【 0 0 0 9 】

また、他の 1 つは、ある命令を委託元マシン 1 から委託先マシン 2 に委託したとしても、前述するように当該命令内において委託先マシン 2 が委託元マシン 1 に別の命令を委託するといった、命令の入れ子が生じたとき、別の命令を行う委託元マシン 1 0 0 のスレッドが排他対象となってしまう、委託元マシン 1 側で排他管理ができなくなる問題がある。

【 0 0 1 0 】

本発明は、上記事情にかんがみてなされたものであって、処理の分散化により発生するデッドロックを確実に回避する自動分散処理システムおよび記録媒体を提供することを目的とする。

【 0 0 1 1 】

【課題を解決するための手段】

(1) 上記課題を解決するために、本発明に係る自動分散処理システムは、命令委託元マシンと命令委託先マシンがネットワークを介して接続されている自動分散処理システムにおいて、



前記命令委託元マシンは、スレッド識別子のもとに各スレッドを管理するテーブルと、委託元上位ライブラリとともに委託元アプリケーションの処理過程における命令発生時、前記テーブルで管理するスレッド識別子を付加して命令を前記委託先マシンに送信する委託元命令中継スレッドと、前記委託先マシンからの別命令を処理するためのスレッドの振り分けを行う委託元命令振り分けスレッドとを有する命令中継ライブラリを設け、また前記命令委託先マシンは、前記委託元命令中継スレッドから送信されてくる命令をスレッド識別子とともに受信し、命令を処理するスレッドを作成し前記スレッド識別子とともに渡す委託先命令振り分けスレッドと、この受け取った命令を委託先上位ライブラリとともに処理し、その命令処理時の別命令発生や処理終了時、前記スレッド識別子を付加して前記別命令や処理終了返信を前記委託元命令振り分けスレッドに送信する命令処理スレッドとを有する命令実行モジュールを設けた構成である。

## 【 0 0 1 2 】

この発明は以上のような構成とすることにより、委託元マシン側のアプリケーション処理によって命令が発生したとき、その命令にスレッド識別子を付けて委託先マシンの命令振り分けスレッドに送信し受信待ちとなる。この命令および識別子を受信した命令振り分けスレッドは、命令処理スレッドを作成し、識別子と共に命令を渡す。

## 【 0 0 1 3 】

この命令処理スレッドは、命令処理後や命令処理時の新たな別命令の発生時、それら命令処理終了や別命令に前記識別子を付けて委託元マシンのスレッド振り分けスレッドに送信する。そこで、委託元マシンのスレッド振り分けスレッドはその識別子から命令発行元となった命令中継スレッドに渡すので、当該命令中継スレッドでは、受信待ちの状態から命令終了や別命令を処理できる状態に移行し処理するので、別命令が発生してもデッドロックを容易に回避することが可能となる。

## 【 0 0 1 4 】

なお、以上の一連の処理は、委託先マシンから命令が発生したときにも同様の構成を持たせることにより、実現可能である。この場合には委託先マシンが委託

元マシンと同様な構成をもてばよく、一方、委託元マシンが委託先マシンと同様な構成を持たせればよい。

【0015】

また、委託元マシンおよび委託先マシンは、予め処理手順を規定するプログラムを記録媒体に記録しておけば、そのプログラムを読み取って以上のような一連の処理を実現することも可能である。

【0016】

(2) 本発明に係る自動分散処理システムは、命令委託元マシンおよび命令委託先マシンには、それぞれ排他用ロック後に自身のアプリケーション処理時に命令が発生したとき、ロックをかけて相手側マシンに命令を中継する命令中継スレッドおよびこの命令中継スレッドからの命令を受信し処理する命令処理スレッドが設けられ、そのうち少なくとも前記委託先マシン側命令処理スレッドは、前記委託元マシン側の命令を受信後、自身のマシンにてロック取得可能か否かを判断し、ロック取得不可の場合には前記委託元マシンにリトライを通知する手段と、ロック取得可能な場合にはロックを取得し命令処理終了後に返信し当該ロックを解放する手段とを有し、また、少なくとも前記委託元マシン側命令中継スレッドは、前記委託先マシン側からリトライ通知を受けたとき、前記ロックを一時的に解放した後再び取得して再度命令を中継するリトライを実施する手段と、前記委託元マシン側から命令終了の返信を受けたとき、前記ロックを解放する手段とを備えた構成である。

【0017】

この発明は、以上のような構成とすることにより、委託元マシンは、排他用ロックをかけて命令を委託先マシンに送信すると、委託先マシン側では、命令受信後に自身側におけるロック取得可能か否かを判断し、既に委託先マシン側においてロックがかかっている場合には、ロック取得不可となるので、委託元マシンにリトライを通知する。

【0018】

通知を受けた委託元マシンは、ロックを解放し、再度ロックの再取得による命令中継をリトライするので、両マシンがほぼ同時に命令が発生しても、デッドロ

ックを容易に回避可能である。

【 0 0 1 9 】

両マシンは、同様な構成とすることにより、何れのマシンでも空いてマシンから命令を受信したとき、ロック取得可能か否かを判断することにより、リトライの通知か命令処理を行うことが可能であり、デッドロックの発生を回避可能となる。

【 0 0 2 0 】

また、以上のような委託元マシンおよび委託先マシンは、予め処理手順を規定するプログラムを記録媒体に記録しておけば、そのプログラムを読み取って以上のような一連の処理を実現することも可能である。

【 0 0 2 1 】

(3) また、本発明に係る自動分散処理システムは、命令委託元マシンは、排他用ロック後のアプリケーション処理時に第1の命令が発生したとき、その命令内容に応じて前記ロックを解放し前記第1の命令を前記委託先マシンの命令処理スレッドに送信する手段およびこの命令処理スレッドにおける命令処理の終了を受けてロックを再び取得し、第1の命令を終了する手段とを有する委託元命令中継スレッドと、前記委託先マシンのイベント処理スレッドから送信されてくる第2の命令を処理する委託元命令処理スレッドとを備え、命令委託先マシンは、前記命令中継スレッドから第1の命令を受信したとき、排他用ロックを行って当該第1の命令を処理し、ロックを解放した上でイベント処理スレッドからの再開依頼がなされるまで待機し、再開後、その命令処理の終了後にロックを解放し、前記命令処理終了を前記委託元命令中継スレッドに委託し、待機中の前記命令委託先マシンに再開依頼を通知する前記委託先命令処理スレッドと、排他用ロックを行って自身のイベント処理時に第2の命令が発生したとき、前記委託元命令処理スレッドに委託する前記委託先イベント処理スレッドとを備えた構成とすることもできる。

【 0 0 2 2 】

この発明は以上のような構成とすることにより、命令の内容によって例えばダイアログ表示命令などのときはロックを解放し、命令を相手マシンに中継するの

で、相手マシンによるダイアログ処理終了の受信後にロックを取得し命令を終了させることができる。

【 0 0 2 3 】

【発明の実施の形態】

以下、本発明の一実施の形態について図面を参照して説明する。

【 0 0 2 4 】

図 1 は本発明に係る G U I 等の処理を行う自動分散処理システムの一実施の形態を示す構成図である。

【 0 0 2 5 】

このシステムは、例えば G U I の表示処理部分等を委託する委託元マシン 1 と、この委託元マシン 1 から委託される表示処理を実行する委託先マシン 2 によって構成されている。

【 0 0 2 6 】

この委託元マシン 1 は、 G U I 等に関する一連の処理を規定するアプリケーション 1 1 と、このアプリケーション 1 1 とリンクして動作する上位ライブラリ 1 2 と、この上位ライブラリ 1 2 からの命令を委託先マシン 2 に中継する命令中継ライブラリ 1 3 とからなり、一方、委託先マシン 2 は、命令中継ライブラリ 1 3 からの命令を処理する命令実行モジュール 2 1 と、この命令実行モジュール 2 1 とリンクして動作する前記委託元上位ライブラリ 1 2 との互換性をもっている上位ライブラリ 2 2 と、この上位ライブラリ 2 2 とリンクして動作する下位ライブラリ 2 3 とからなる。

【 0 0 2 7 】

なお、委託元マシン 1 および委託先マシン 2 にはそれぞれデッドロック防止のためのプログラムを記録した記録媒体 1 4 , 2 4 が設けられている。この記録媒体 1 4 , 2 4 は、一般的には C D－R O M や磁気ディスク等が用いられるが、それ以外にも例えば磁気テープ、 D V D－R O M、フロッピーディスク、 M O、 C D－R、メモリカードなどを用いてもよい。

【 0 0 2 8 】

この委託元マシン 1 の命令中継ライブラリ 1 3 は、委託元下位ライブラリ 2 3

と互換性を有し、具体的には上位ライブラリ 12 からの命令を命令実行モジュール 21 に中継する命令中継スレッド 131、委託先マシン側の命令実行モジュール 21 からの命令を処理するスレッドを検索し命令を振り分ける命令振り分けスレッド 132、この命令振り分けスレッド 132 からの命令を上位ライブラリ 12 に渡して処理する命令処理スレッド 133、スレッド識別子を用いてスレッドを管理するスレッド管理テーブル 134、処理が終了し解放された命令処理スレッド 133 を管理する解放済スレッド蓄積部 135 等によって構成されている。

## 【0029】

前記委託先マシン 2 の命令実行モジュール 21 は、委託先下位ライブラリ 22 からの命令を命令中継ライブラリ 13 に中継する命令中継スレッド 211、委託元命令中継ライブラリ 13 からの命令を処理するスレッドを検索し命令を振り分ける振り分けスレッド 212、この振り分けスレッド 212 からの命令を上位ライブラリ 22 に渡して処理する命令処理スレッド 213、スレッド識別子を用いてスレッドを管理するスレッド管理テーブル 214、処理が終了し解放された委託先命令スレッド 213 を管理する解放済スレッド蓄積部 215 等によって構成されている。

## 【0030】

図 2 は一般的にアプリケーションを実行する場合のモジュールの概略構成図である。

## 【0031】

すなわち、実行マシン 100 は、アプリケーション 110、このアプリケーション 110 とリンクして動作する上位ライブラリ 120 と、この上位ライブラリ 120 とリンクして動作する下位ライブラリ 130 とを有し、当該アプリケーション 110 を実行するが、例えば GUI 等の処理を行なうアプリケーション 11 を実行する場合、下位ライブラリ 130 部分を、当該下位ライブラリ 130 と互換性のある命令中継ライブラリ 13 に入れ替え、処理の自動分散化を図っている。

## 【0032】

次に、以上のような自動分散処理システムに関し、特にデッドロック防止の一

連の処理について図 3 を参照して説明する。なお、委託元命令中継スレッド 1 3 1 と委託先命令中継スレッド 2 1 1、委託元命令振り分けスレッド 1 3 2 と委託先命令振り分けスレッド 2 1 2 とはそれぞれ同様な処理を行なうので、説明の便宜上、片方の処理フローは省略する。

#### 【 0 0 3 3 】

命令中継ライブラリ 1 3 は上位ライブラリ 1 2 0 から命令を受けると、記録媒体 1 4 に記録されるプログラムを読み取り、次のような処理を実行する。

#### 【 0 0 3 4 】

今、委託元マシン 1 においてアプリケーション 1 1 の実行時に上位ライブラリ 1 2 を通して命令が発生すると (S 1)、ユーザスレッド等の命令中継スレッド 1 3 1 は、その命令処理用スレッドがスレッド管理テーブル 1 3 4 内に登録されているか否かを調べ (S 2)、登録されていない場合には当該スレッド管理テーブル 1 3 4 に登録し (S 3)、次のステップ S 4 に移行する。既にスレッドが登録されている場合にも同様にステップ S 4 に移行する。このステップ S 4 では、命令に発行元のスレッド識別子を付加し、委託先マシン 2 の命令振り分けスレッド 2 1 2 に中継するとともに、自身を受信待ちの状態に設定する。これらステップ S 1 ~ S 4 は命令を中継する機能である。

#### 【 0 0 3 5 】

ここで、委託先マシン 2 の命令振り分けスレッド 2 1 2 は、命令を処理するスレッド側のスレッド識別子がスレッド管理テーブル 2 1 4 に登録されているか否かを判断し (S 5 1)、登録されていなければ解放済スレッド蓄積部 2 1 5 にスレッドが有るか否かを調べ (S 5 2)、スレッドが無ければ新たに識別子とともにスレッドを作成し (S 5 3)、スレッド識別子と命令処理スレッドとを組としてスレッド管理テーブル 2 1 4 に登録後 (S 5 4)、その命令処理スレッド 2 1 3 に命令を渡す (S 5 5)。ステップ S 5 2 において解放済スレッドがある場合もステップ S 5 4 を介して命令処理スレッドに命令を渡す。これらステップ S 5 1 ~ S 5 5 は命令処理用スレッドを作成し命令を渡す機能である。

#### 【 0 0 3 6 】

命令処理スレッド 2 1 3 は、上位ライブラリ 2 2 とリンクして処理動作を実行

するが、このとき処理が終了する場合と処理の過程で別命令が発生する場合があるので、別命令発生か否かを判断し（S56）、別命令発生の場合には前述した委託元マシン1から受信した識別子を付けて別命令を委託元マシン1の命令振り分けスレッド132に送出する。この時、委託先マシン側の識別子を付けて送信してもよい。また、別命令の有無を判断し、別命令無しの場合には上位ライブラリ22とリンクしながら該当命令の処理が終了したかを判断し（S57）、処理終了の場合には該当命令終了の返信に同じく委託元マシン1から受け取った識別子を付けて委託元マシン1の命令振り分けスレッド132に送出する（S58）。

## 【0037】

委託元マシン1の命令振り分けスレッド132は、識別子とともに委託先の別命令または処理終了による返信を受け取ると、識別子から委託元である命令中継スレッド131に処理を委託する。この命令中継スレッド131は、命令に対する返信か、別命令かを判断（S6）、返信の場合には終了し、別命令の場合には上位ライブラリ12に別命令を発行し、処理を行なう（S7）。ここで、別命令の処理が終了すれば、別命令に対する処理終了の返信を委託先側発行元のスレッド識別子とともに委託先マシン2の命令振り分けスレッド212に送信する（S8）。

## 【0038】

従って、以上のような実施の形態によれば、デッドロックを容易に回避可能である。

## 【0039】

すなわち、ある命令が委託元マシン1から委託先マシン2に行われ、この命令内で呼ばれる委託先上位ライブラリ22内から更に別命令が委託元マシン1に委託されると、図9に示すごとく命令の入れ子が行われ、デッドロックが発生する。

## 【0040】

しかし、本実施の形態では、委託先で別の命令が発生したとき、委託元から既に命令委託のときに受け取っているスレッド識別子を別命令とともに返すので、

委託元マシンでは委託元のスレッドが判明でき、この委託元である命令中継スレッド 1 3 1 はロットをかけているが、命令に伴う別名例であることがスレッド識別子からわかるので、処理の自動分散化を行なわない場合と同様に確実にデッドロックを回避できる。

#### 【 0 0 4 1 】

図 4 は図 9 とは異なる別のデッドロック発生例を説明するための図である。

#### 【 0 0 4 2 】

このシステムは、通常のアプリケーション実行時における委託元マシン 1 のユーザスレッド等の命令中継スレッド 1 3 1 と委託先マシン 2 のイベント処理スレッド（命令中継スレッド） 2 1 1 とで同一の排他用ロックを同時に取得しようとするアプリケーションの例である。

#### 【 0 0 4 3 】

この場合、命令中継スレッド 1 3 1 がロックをかけて委託元アプリケーションの処理時に命令 A が発生したとき、その命令 A を委託先マシン 2 の命令処理スレッド 2 1 3 に中継し、処理待ちの状態に入る。一方、イベント処理スレッド 2 1 1 においても、ロックをかけた状態でのイベント処理時に命令 B が発生したとき、その命令 B を委託元マシン 1 の命令処理スレッド 1 3 3 に中継し、処理待ちの状態に入る。しかし、これらマシン 1, 2 はそれぞれ時間をずらして各命令を相手マシン 2, 1 に命令を委託する場合には特に問題が生じないが、両マシン 1, 2 が同時に命令が発生した場合には互いにロックを取得し合えない状態となり、デッドロックが発生する。

#### 【 0 0 4 4 】

図 5 は図 4 に示すデッドロックを回避するための本発明に係る自動分散処理システムの他の実施形態を示す構成図である。

#### 【 0 0 4 5 】

今、委託元マシン 1 において命令中継スレッド 1 3 1 は、ロックをかけて委託元アプリケーション処理時に命令 A が発生すると（S 1 1）、命令 A を中継し（S 1 2）、委託先マシン 2 の命令処理スレッド 2 1 2 に送信する。このステップ S 1 1, S 1 2 は命令中継機能である。



## 【0046】

この命令処理スレッド212は、命令Aを受信すると（S61）、自身のマシンに対するロックが取得可能か否かを判断する（S62）。ステップS61、S62はロック取得判断機能である。ロック取得不可であれば、委託元マシン1にリトライするよう通知する（S63：リトライ通知機能）。ロックを取得可能であれば、ロックをかけて命令Aを処理し（S64）、命令処理終了後にその命令Aを終了した旨を返信し（S65）、ロックを解放する。ステップS64、S65は命令処理機能である。

## 【0047】

一方、委託元マシン1の命令中継スレッド131では、命令Aを中継した後（S12）、受信待ちの状態に入る（S13）。命令A中継後、リトライ用の通知があったとき（S14）、ロックを一時解放し（S15）、ロックの再取得を命令A中継に指示する（S16）。これらステップS13～S16は命令中継リトライ機能である。

## 【0048】

また、命令中継スレッド131では、委託先マシン2の命令処理スレッド212からの命令処理返信の有無を判断し（S17）、処理終了であればロックを解放し、命令処理を終了する（S18）。

## 【0049】

なお、委託先マシン2によるイベント処理時に命令Bが発生したとき、命令Bを中継し、委託元マシン1の命令処理スレッド132に送信する。この命令処理スレッド132は、ロックを取得し命令Bを処理し、命令処理終了後に返信する構成とするが、例えば委託先マシン2の命令処理スレッド212と同様な構成であってもよい。また、委託先マシン2の命令中継スレッド211についても、委託元マシン1の命令中継スレッド131と同様な構成であってもよい。

## 【0050】

従って、以上のような実施の形態によれば、委託先マシン2が委託元マシン1から命令を受信したとき、自身マシンのロックを取得できるか判断し、ロック取得不可の場合に委託元マシン1のリトライを通知し、再度当該命令の中継を促す

ので、デッドロックを容易に回避できる。

【0051】

一般に、デッドロックの回避手段としては、最初の命令を中継するとき、排他用ロックを無条件に解放することもあるが、もともとアプリケーションが排他用ロック取得状態にて実行している命令内であるので、不用意に排他用ロックを解放すると、排他機能が働かず、予期せぬ不適合が発生する可能性がある。

【0052】

そこで、本システムにおいては、委託元マシン1上の排他機能をできる限り生かし、例外的にデッドロックの危険性が有るときだけ、デッドロックを回避するものである。

【0053】

次に、図6および図7にて更に別のデッドロックの発生例について述べる。

【0054】

例えば同一マシン上、つまり委託元マシン1または委託先マシン2において処理画面上のあるアイコンをマウス等でクリックしダイアログ表示命令を出し、マウスイベント等に伴うダイアログボックス（イベント処理スレッド）を生成したとき、そのダイアログボックスが閉じるまで待機状態にある場合のデッドロック発生例である。

【0055】

まず、入力イベント待ちダイアログ表示の際の実行マシン100の通常実行動作は、図6に示すようにダイアログ表示命令が発生する（S21）、排他用ロックを取得し、ダイアログを表示した後（S22）、そのダイアログ表示の旨をイベント処理スレッドに通知する（S23）。この通知後、排他用ロックを解放し（S24）、待機状態とする（S25）。つまり、一度ロックを取得するが、ダイアログ表示の旨を通知した後、ロックを解放し、待機状態にする。

【0056】

一方、イベント処理スレッドでは、ダイアログ表示の通知を受信すると、ダイアログ表示通知有りと判断し（S71）、ダイアログ管理部に登録するが（S72）、ダイアログ表示通知無しの場合には、管理中ダイアログの非表示イベント

であるか否かを判断し（S 7 3）、非表示の場合にはユーザスレッドの待機中スレッドを起こし（S 7 4）、ダイアログの非表示イベントでない場合にはロックを取得し、イベント処理を行ない、その処理終了後にロックを解放する（S 7 5）。

## 【 0 0 5 7 】

ステップ S 7 4 においてダイアログ非表示イベントに入った際、待機中に合ったスレッドを再開し、ロックを取得する（S 2 6）。

## 【 0 0 5 8 】

従って、以上のような同一マシン上での通常実行時、互いに相手のスレッドを調べながらロックをかけたり、ロックを解放しているので、デッドロックの発生がない。

## 【 0 0 5 9 】

しかしながら、以上のようなダイアログの表示技術に関し、本発明に係る自動分散処理システムに適用すると、図 7 に示すようにデッドロックが発生する。

## 【 0 0 6 0 】

すなわち、委託元マシン 1 における委託元アプリケーションの処理時、命令中継スレッド 1 3 1 は、ダイアログ表示命令 A 発生後、ロックをかけた後、命令 A を委託先マシン 2 の命令処理スレッド 2 1 3 に送信した後、命令処理待ちの状態に入る。

## 【 0 0 6 1 】

一方、委託先マシン 2 の命令処理スレッド 2 1 3 は、命令 A を受信した後、命令 A を処理する。つまり、ロックをかけて図 6 のユーザスレッドの処理を行ない、ロック解放、待機を行なう。

## 【 0 0 6 2 】

ところが、委託先マシン 2 の命令中継スレッド 2 1 1 においては、同様にイベント処理時に命令 B が発生したとき、命令 B を委託元マシン 1 の命令処理スレッド 1 3 3 に送信するので、命令 B を受信した命令処理スレッド 1 3 3 では、命令 B を受信するが、ロックがかかった状態にあるので、デッドロックが発生する。つまり、委託先マシン 2 上のイベント処理スレッド 2 1 1 のイベント処理中に排

他用ロックを取得しようとした場合、イベント処理スレッドの動作が停止してしまい、デッドロックが発生する。

## 【0063】

そこで、本発明システムでは、委託元マシン1の命令中継スレッド131は、命令A（ダイアログ表示命令）が発生したとき（S31）、ロックをかけるが、その命令Aがダイアログ表示命令であれば（S32）、ロックを解放し（S33）、命令Aを委託先マシン2の命令処理スレッド213に中継する（S34）。ステップS31～S33は命令解析ロック解除機能である。

## 【0064】

委託先マシン2の命令処理スレッド213は、命令Aを受信するとその命令内容であるダイアログ表示に関し、ロックをかけて処理し（処理内ではロックを解放して待機し、再開依頼を待つ）、その処理終了後に返信する（S81）。ここで、委託元マシン1の命令中継スレッド131は、委託先マシン2からの返信を受信し（S35）、命令Aを終了する（S36）。

## 【0065】

一方、委託先マシン2の命令中継スレッド211は、所定周期でダイアログ非表示イベント発生の有無を判断し（S82）、ダイアログ非表示時にはイベント処理（ロック取得）を実行し命令Bが発生したとき（S83）、委託元マシン1の命令処理スレッド133に送信し、命令処理後の返信を受けて命令Bを終了し（S84）、ロックを解放する。

## 【0066】

従って、以上のような実施の形態によれば、委託元マシン1の命令中継スレッド131は、命令A発生時、ロックをかけるが、ダイアログ表示命令が発生したとき、つまり委託先の処理を待つ必要がある命令の場合には、ロックを解放し、委託先マシン2に中継するので、委託先マシン2のイベント処理スレッド211がイベント処理の過程で新たな命令Bが発生し委託元マシン2のロックが解除されているので、命令Bが発生しても委託元マシン1はデッドロックを容易に回避できる。

## 【0067】

なお、本願発明は、上記実施の形態に限定されるものでなく、その要旨を逸脱しない範囲で種々変形して実施できる。また、各実施の形態は可能な限り組み合わせさせて実施することが可能であり、その場合には組み合わせによる効果が得られる。さらに、上記各実施の形態には種々の上位、下位段階の発明が含まれており、開示された複数の構成要素の適宜な組み合わせにより種々の発明が抽出され得る。例えば実施の形態に示される全構成要件から幾つかの構成要件が省略されることで発明が抽出された場合には、その抽出された発明を実施する場合には省略部分が周知慣用技術で適宜補われるものである。

【0068】

【発明の効果】

以上説明したように本発明によれば、命令委託先の処理部をそのままの形で使用しながらデッドロックを未然に防止できる。

【0069】

また、ある命令が委託元マシンから委託先マシンへ行われ、当該命令内で呼ばれる委託先マシンの処理部内から更に委託元マシンに別命令を行なう、いわゆる命令の入れ子が行われた場合でも、処理の分散処理を行なわない場合と同様にデッドロックを防止でき、また逆に委託先マシンから委元マシンへの委託であっても同様である。

【0070】

さらに、通常実行時は同一であるはずの排他用ロックを、委託元マシンと委託先マシンとで同時に取得した場合でも、委託先マシンおよび委託元マシンの排他用ロックの取得可否のみにより排他を行なうことにより、容易にデッドロックを防止できる。

【0071】

さらに、通常実行時、排他的ロックを取得して処理を行なった後、排他的ロックを解放し、その後、特定の命令が行われるまで命令発行元スレッドが待機するといった性質をもつ命令を行なった場合のデッドロックも確実に防止できる。

【図面の簡単な説明】

【図1】 本発明に係る自動分散処理システムの一実施の形態を示す構成図

【図 2】 アプリケーションの通常実行時のモジュールの概略構成図。

【図 3】 デッドロック防止処理を説明するフローチャート。

【図 4】 デッドロックの発生する例を説明する図。

【図 5】 図 4 に示すデッドロックを回避する本発明に係る自動分散処理システムの他の実施形態を示す構成図。

【図 6】 実行マシンの通常実行時の処理を説明するフローチャート。

【図 7】 デッドロックの発生する他の例を説明する図。

【図 8】 図 7 に示すデッドロックを回避する本発明に係る自動分散処理システムの更に他の実施形態を示す構成図。

【図 9】 従来一般的なデッドロック発生例を説明する図。

【符号の説明】

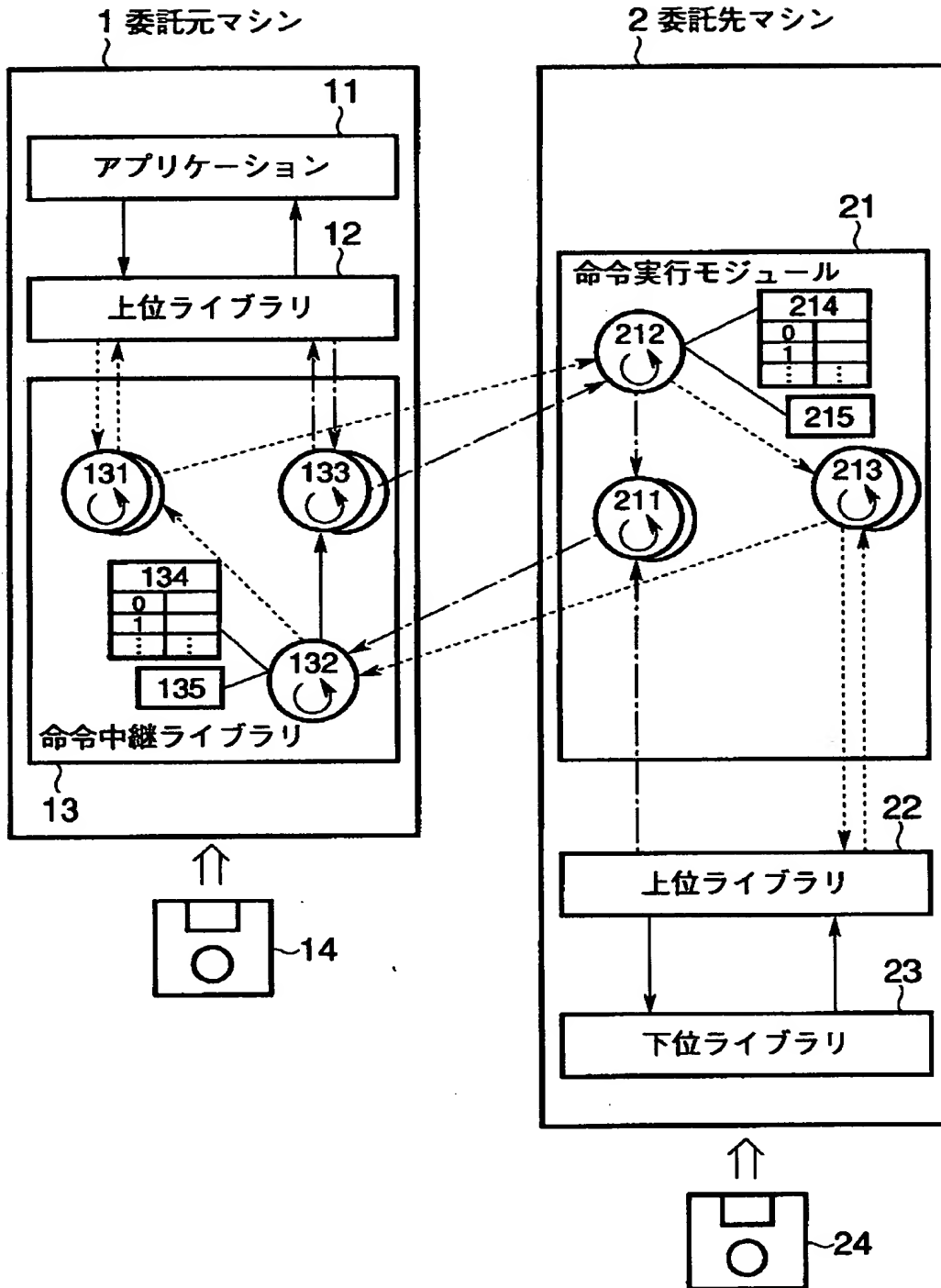
- 1 …委託元マシン
- 2 …委託先マシン
- 1 1 …アプリケーション
- 1 2 …上位ライブラリ
- 1 3 …命令中継ライブラリ
- 1 4 …記録媒体
- 2 1 …命令実行モジュール
- 2 2 …上位ライブラリ
- 2 4 …記録媒体
- 1 3 1 …命令中継スレッド
- 1 3 2 …命令振り分けスレッド
- 1 3 3 …命令処理スレッド
- 1 3 4 …スレッド管理テーブル
- 1 3 5 …解放済スレッド蓄積部
- 2 1 1 …命令中継スレッド
- 2 1 2 …命令振り分けスレッド
- 2 1 3 …命令処理スレッド

2 1 4 …スレッド管理テーブル

2 1 5 …解放済スレッド蓄積部

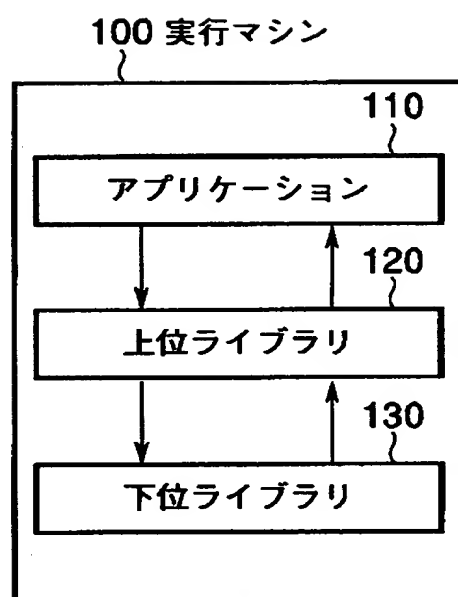
【書類名】 図面

【図 1】

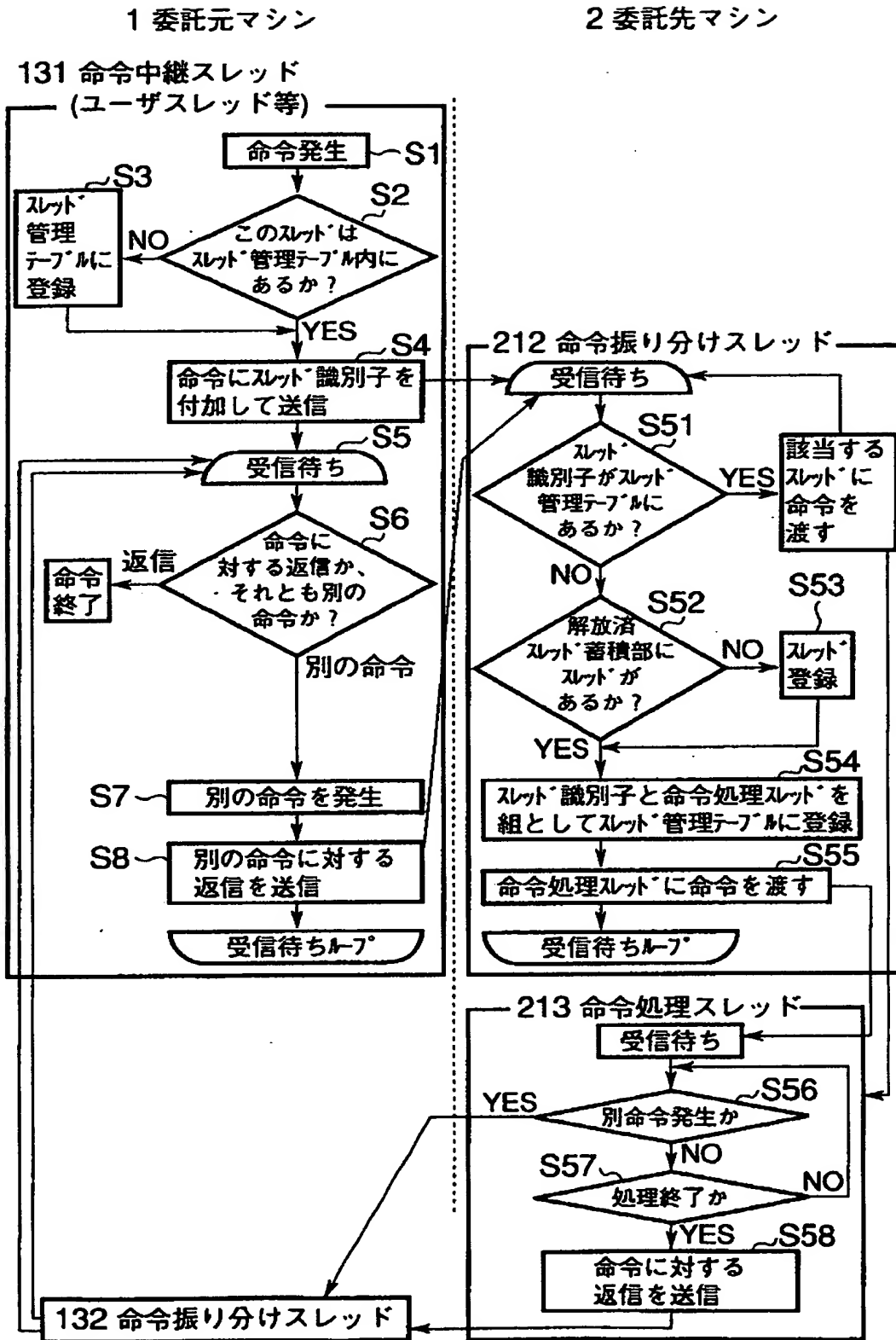




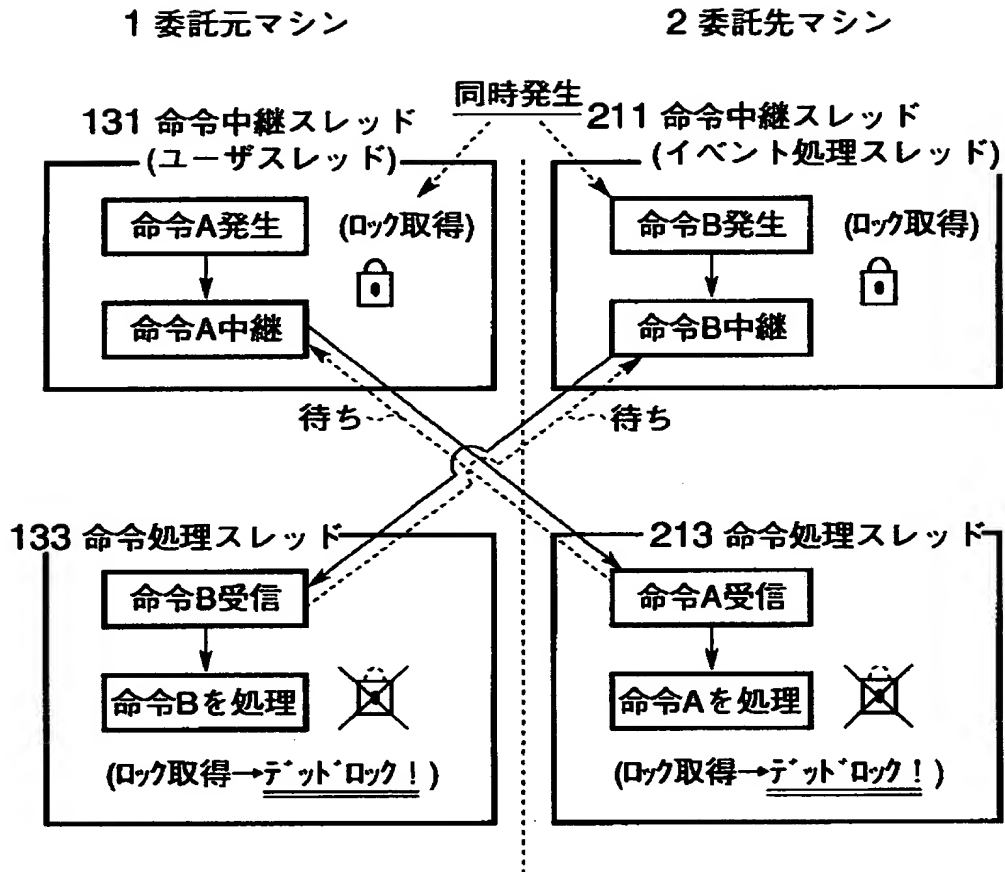
【図 2】



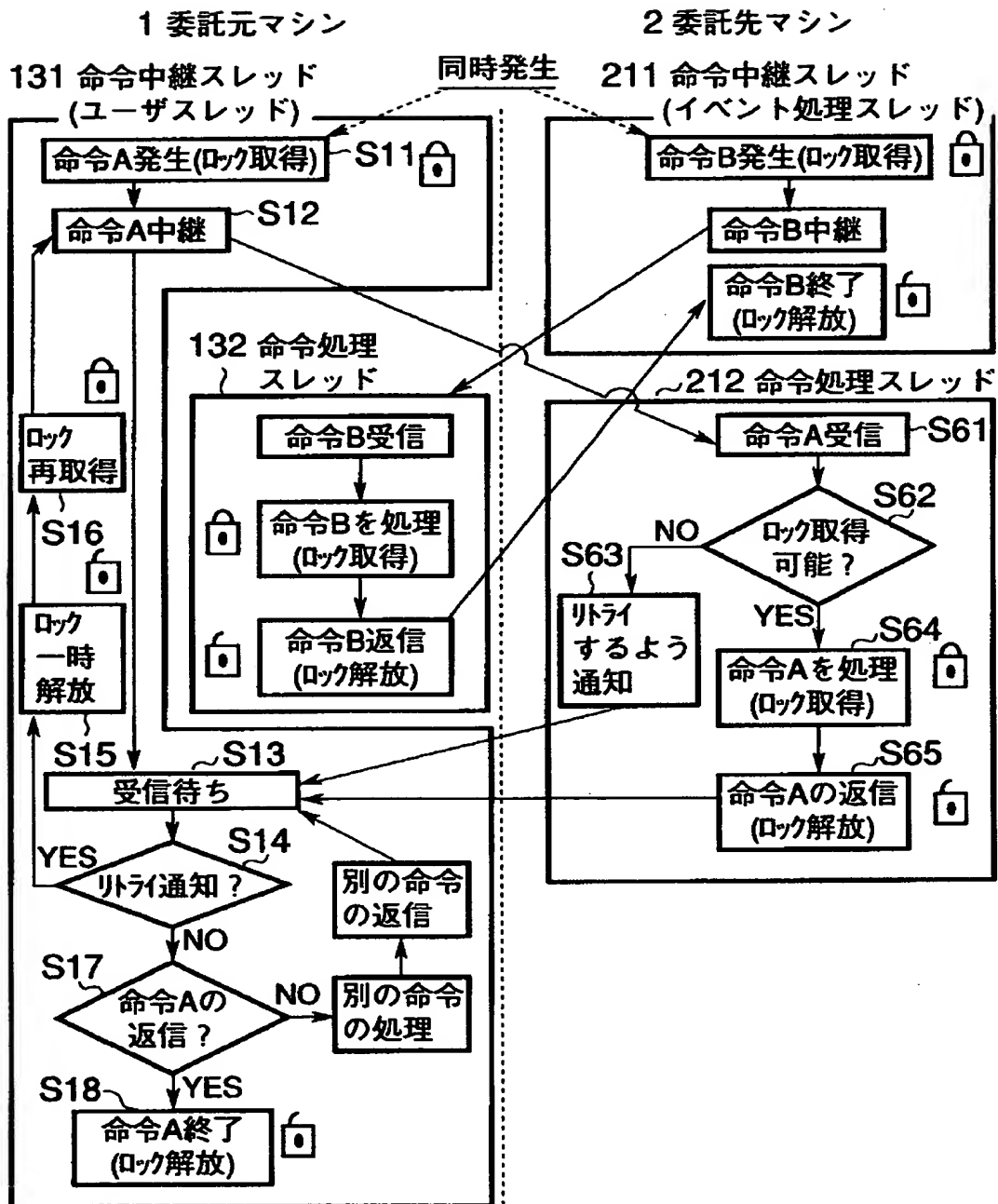
【図 3】



【図 4】

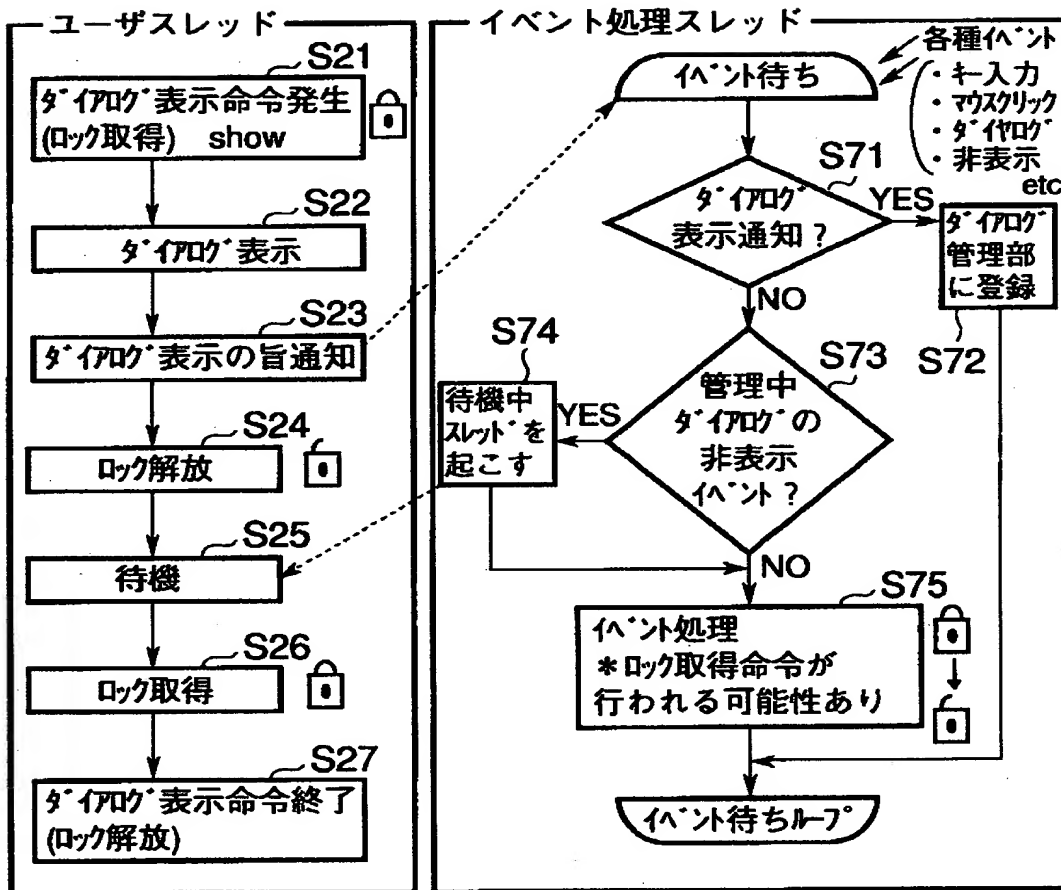


【図5】

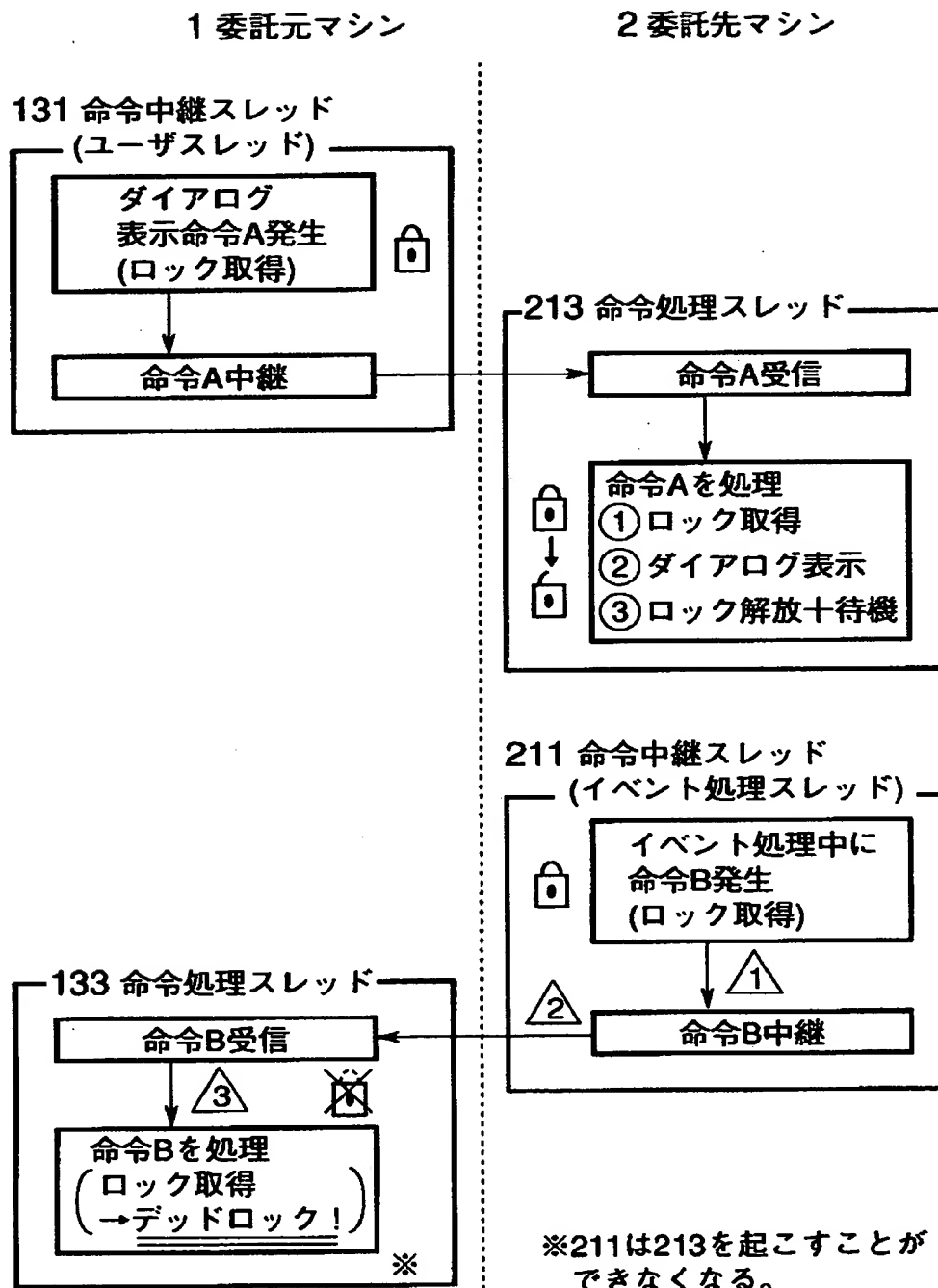


【図6】

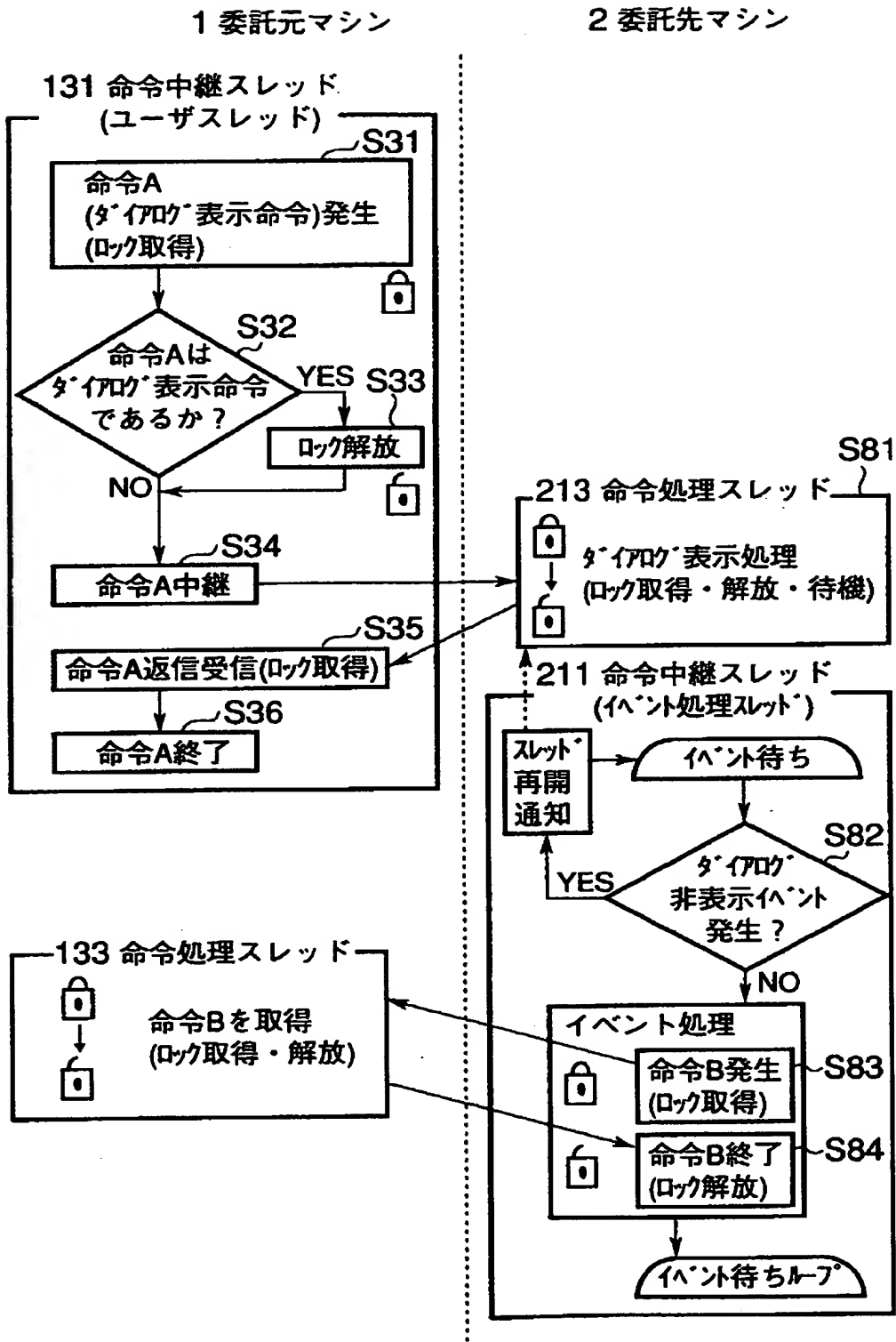
100 実行マシン



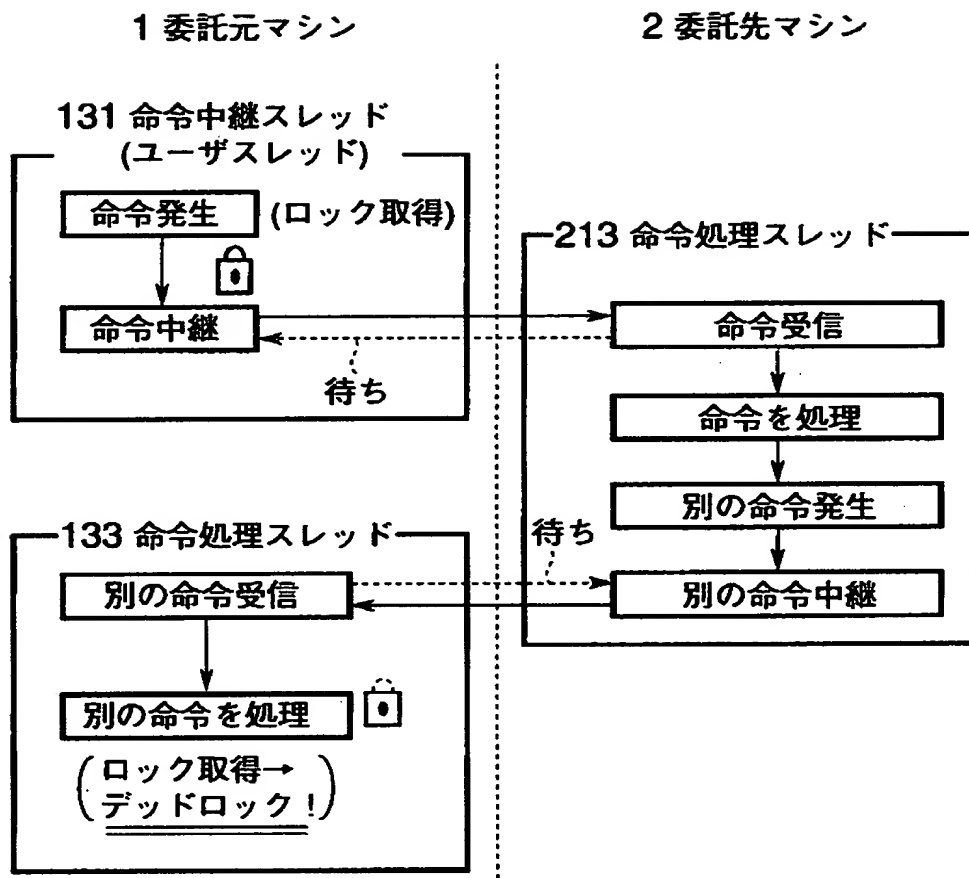
【図 7】



【図8】



【図 9】





【書類名】 要約書

【要約】

【課題】 分散処理時のデッドロックを防止することにある。

【解決手段】 命令委託元マシン1と命令委託先マシン2がネットワークを介して接続され、そのうち前記命令委託元マシンは、委託元上位ライブラリ12とともに委託元アプリケーション11の処理過程における命令発生時、スレッド識別子を付加して命令を中継する委託元命令中継スレッド131と、委託先マシンからの別命令を処理するスレッドを検索する委託元命令振り分けスレッド132とを有する委託元中継ライブラ13設け、前記命令委託先マシンは、前記スレッド識別子とともに命令を受信し、命令を処理するスレッドを作成し前記スレッド識別子とともに渡す委託先命令振り分けスレッド212と、この受け取った命令を委託先上位ライブラリ22とともに処理し、その命令処理時の別命令発生や処理終了時、前記スレッド識別子を付加して別命令等を前記委託元命令振り分けスレッドに送信する命令処理スレッド213とを有する命令実行モジュールとを備えた自動分散処理システムである。

【選択図】 図3

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 3 0 7 8 ]

1. 変更年月日	1 9 9 0 年 8 月 2 2 日
[変更理由]	新規登録
住 所	神奈川県川崎市幸区堀川町 7 2 番地
氏 名	株式会社東芝